REMARKS/ARGUMENTS

Rejections 35 U.S.C. §103

Claims 1-3, 5-7, 9-12 and 14 are rejected, under 35 U.S.C. §103(a), as being allegedly unpatentable by Larsen et al, (US Pat No. 5,115,500) (hereinafter Larsen). Applicants respectfully traverse in view of the following.

Independent Claim 1 recites fetching an instruction using a corresponding address from a memory unit, wherein a plurality of possible meanings are associated with the instruction stored at the corresponding address by a same processor, as claimed. As a result, the same instruction stored in the same memory location is interpreted differently and given different meanings such that it can have a plurality of possible meanings. For example, concatenating the least significant bit of an address with the instruction produces one meaning whereas concatenating the least two significant bits of the same address with the same instruction produces a different meaning (see Instant Application, Figure 1, elements 10, 20, 30 and 40). Independent Claim 1 further recites concatenating a portion of the corresponding address to the instruction to form an extended instruction, as claimed.

In contrast, Larsen discloses storing incompatible machine languages for two or more different machine types, yet allowing the incompatible languages to

be properly decoded (see Larsen, col. 4, lines 58-64) by storing incompatible languages in various portions of segregated locations of the I-store (see Larsen, col. 5, lines 4-7). For example, an instruction identified by the three high order address bits "111" are decoded according to "type 2" decode rule whereas other address bits are decoded according to "type 1" (see Larsen, col. 6, lines 30-40).

Accordingly, the instruction identified by the three high order address bits "111" is decoded differently than if the same instruction was identified by the three high order address bits other than "111." As a result, to have two different decodings for the same instruction requires the same instruction to be stored in two different locations, e.g., identified by the three high order address bits "111" and other than "111." In other words, an instruction stored in a location has only one type. Thus, in order to associate the instruction with a different type requires the instruction to be stored in a different location of a memory component. As such, Larsen fails to teach or suggest fetching an instruction using a corresponding address from a memory unit, wherein a plurality of possible meanings are associated with the instruction stored at the corresponding address by a single processor, as claimed.

Moreover, an instruction being stored in various memory locations to have various machine types, as disclosed by Larsen, differs from a plurality of meanings associated with the instruction stored at the corresponding address, as

claimed. For example, operations of the same sort written in different format, e.g., "type 2 add registers", is to be executed (see Larsen, col. 7, lines 38-40). The instruction for "add registers" for type 2 is decoded based on the binary content "111" of the three high order bits, which is "XX1A32" (see Larsen, col. 7, lines 45-53). The execution code to be placed in to execute register is identical since exactly the same function is specified by both type 1 add register's instruction and the type 2 add register's instruction (see Larsen, col. 8, lines 10-13). As a result, the state of the execute register after the type 2 add register's instruction has been successfully decoded as being the same as it was for the type 1 instruction (see Larsen, col. 8, lines 13-18).

Thus, Larsen discloses the same instruction, e.g., add instruction, written in two different formats, e.g., type 1 and type 2. An add instruction written in different formats has the same meaning, e.g., adding two things. Thus, different formats, e.g., type 1 and type 2, as disclosed by Larsen, fails to teach or suggest a plurality of meanings associated with the instruction stored at the corresponding address, as claimed.

As presented and discussed above, Larsen discloses using the instruction with the three high order address bits. An instruction stored at a given address, as disclosed by Larsen, has only one meaning and one format. For example, using the three high order address bits "111" with a given instruction, e.g., add,

always corresponds to an add instruction of type 2 format. Thus, concatenating "111" with the add instruction does not extend the number of instructions, as disclosed by Larsen. The number of instruction, as disclosed by Larsen, is increased by storing the instruction in different locations, hence having a different three high order address bits, e.g., "001." As such, Larsen fails to teach or suggest concatenating a portion of the corresponding address to the instruction to form an extended instruction, wherein the concatenation increases a number of instructions in an instruction set, as claimed.

Accordingly, Larsen fails to render independent Claim 1, under 35 U.S.C. §103(a). Independent Claims 5 and 10 recite limitations similar to that of independent Claim 1 and are patentable for similar reasons. Dependent claims are patentable by virtue of their dependency. As such, allowance of Claims 1-3, 5-7, 9-12 and 14 is earnestly solicited.

As per Claims 9 and 14, the rejection asserts that the Applicants are correct in the statements concerning the function of a compiler. The rejection asserts that the Applicant fails to state "why" Larsen does not read on the limitation. Applicants respectfully reiterate the fact that a mere disclosure of a "compiler," as disclosed by Larsen, does not teach or suggest that it is the compiler that generates the instruction and the act of the storing of the instruction, as claimed. The rejection asserts that the compiler has to inherently

follow the semantic rule of where the different instruction types can be stored. Applicants respectfully disagree because one instruction executable on a first processor may be translated into an instruction executable on a second processor wherein the translation is performed in machine code. Accordingly, no compiler is used to perform the translation. As such, the assertion by the rejection that the compiler has to inherently follow the semantic rule of where the different instruction types can be stored is misplaced.

Moreover, Applicants wish to remind the Examiner that to establish inherency, the extrinsic evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient (see In re Robertson, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999)). In relying upon the theory of inherency, the rejection must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art (see Ex parte Levy, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990)).

Claims 4, 8 and 13 are rejected, under 35 U.S.C. §103(a), as being allegedly unpatentable over Larsen in view of ("390 Principles of Operation") (hereinafter IBM). Applicants respectfully submit that IBM fails to remedy the failures of Larsen with respect to independent Claims 1, 5 and 10. Claims 4, 8 and 13 depend from independent Claims 1, 5 and 10 respectively. Thus, Larsen alone or in combination with IBM fails to render Claims 4, 8 and 13 obvious. As such, allowance of Claims 4, 8 and 13 is earnestly solicited.

For the above reasons, the Applicants request reconsideration and withdrawal of the rejections of record.

CONCLUSION

In light of the above listed remarks, reconsideration of the rejected Claims 1-14 is requested. Based on the arguments presented above, it is respectfully submitted that Claims 1-14 are in condition for allowance.

Please charge any additional fees or apply any credits to our PTO deposit account number: 50-4160.

Respectfully submitted,
MURABITO, HAO & BARNES LLP

Dated: 6/17, 2008

/Michael D. Sochor/
Michael D. Sochor
Registration No. 58,348

MURABITO, HAO & BARNES LLP
Two North Market Street
Third Floor
San Jose, California 95113

(408) 938-9060 Voice
(408) 938-9069 Facsimile